ANT - BUILDING PROJECTS

Now that we have learnt about the data types in Ant, it is time to put that into action. Consider the following project structure

This project will form the **Hello World** Fax Application project for the rest of this tutorial.

```
C:\work\FaxWebApplication>tree
Folder PATH listing
Volume serial number is 00740061 EC1C:ADB1
C:.
+---db
+---src
.   +---faxapp
.       +---dao
.       +---entity
.       +---util
.       +---web
+---war
    +---images
    +---js
    +---META-INF
    +---styles
    +---WEB-INF
        +---classes
        +---jsp
        +---lib
```

Let me explain the project structure.

- The database scripts are stored in the **db** folder.
- The java source code is stored in the **src** folder.
- The images, js, META-INF, styles (css) are stored in the **war** folder.
- The JSPs are stored in the **jsp** folder.
- The third party jar files are stored in the **lib** folder.
- The java class files will be stored in the **WEB-INF\classes** folder.

The aim of this exercise is to build an ant file that compiles the java classes and places them in the WEB-INF\classes folder.

Here is the build.xml required for this project. Let us consider it piece by piece

```
<?xml version="1.0"?>
<project name="fax" basedir="." default="build">
    <property name="src.dir" value="src"/>
    <property name="web.dir" value="war"/>
    <property name="build.dir" value="${web.dir}/WEB-INF/classes"/>
    <property name="name" value="fax"/>

    <path >
        <fileset dir="${web.dir}/WEB-INF/lib">
            <include name="*.jar"/>
        </fileset>
        <pathelement path="${build.dir}"/>
    </path>

    <target name="build" description="Compile source tree java files">
```

```
        <mkdir dir="${build.dir}"/>
        <javac destdir="${build.dir}" source="1.5" target="1.5">
            <src path="${src.dir}"/>
            <classpath ref/>
        </javac>
    </target>

    <target name="clean" description="Clean output directories">
        <delete>
            <fileset dir="${build.dir}">
                <include name="**/*.class"/>
            </fileset>
        </delete>
    </target>
</project>
```

First, let us declare some properties for the source, web and build folders.

```
<property name="src.dir" value="src"/>
<property name="web.dir" value="war"/>
<property name="build.dir" value="${web.dir}/WEB-INF/classes"/>
```

In this example, the **src.dir** refers to the source folder of the project (i.e, where the java source files can be found).

The **web.dir** refers to the web source folder of the project. This is where you can find the JSPs, web.xml, css, javascript and other web related files

Finally, the **build.dir** refers to the output folder of the project compilation.

Properties can refer to other properties. As shown in the above example, the **build.dir** property makes a reference to the **web.dir** property.

In this example, the **src.dir** refers to the source folder of the project.

The default target of our project is the **compile** target. But first let us look at the **clean** target.

The clean target, as the name suggests deletes the files in the build folder.

```
<target name="clean" description="Clean output directories">
   <delete>
      <fileset dir="${build.dir}">
         <include name="**/*.class"/>
      </fileset>
   </delete>
</target>
```

The master-classpath holds the classpath information. In this case, it includes the classes in the build folder and the jar files in the lib folder.

```
<path >
   <fileset dir="${web.dir}/WEB-INF/lib">
      <include name="*.jar"/>
   </fileset>
   <pathelement path="${build.dir}"/>
</path>
```

Finally, the build target to build the files. First of all, we create the build directory if it doesn't exist. Then we execute the javac command (specifying jdk1.5 as our target compilation). We supply the source folder and the classpath to the javac task and ask it to drop the class files in the build folder.

```
<target name="build" description="Compile main source tree java files">
   <mkdir dir="${build.dir}"/>
   <javac destdir="${build.dir}" source="1.5" target="1.5" debug="true"
```

```
              deprecation="false" optimize="false" failonerror="true">
      <src path="${src.dir}"/>
      <classpath ref/>
   </javac>
</target>
```

Running ant on this file will compile the java source files and place the classes in the build folder.

The following outcome is the result of running the ant file:

```
C:\>ant
Buildfile: C:\build.xml

BUILD SUCCESSFUL
Total time: 6.3 seconds
```

The files are compiled and are placed in the **build.dir** folder.